IN THE CLAIMS

1. (Original) A system having improved system management interrupt (SMI) latency, comprising:

a real-time clock;

a register file containing one or more timing sensitive registers;

an index and data register for accessing the timing sensitive registers in the register file;

an update-in-progress status bit that determines a certain fixed period of time for which the timing-sensitive registers are valid;

a retriggerable, fixed duration timer that is triggered by reads of zero from the update-in-progress status bit;

a latch that is set if the timer is running when a system management interrupt is asserted and cleared when SMI is deasserted;

means for reading the output status of the latch;

a timer that is triggerable by reading zero from a first register location;

a status latch for storing the status of the timer, whish status is read using a status bit; and

SMI handling code that reads the status latch, and if the status latch is zero, exits the SMI handling code, and if the status latch is non-zero, writes to a second register location, reads a third register location, and if a predetermined bit of the value read from the third register location is set, repeats the previous two steps until the value of the bit is not set, and then exits the SMI handling code.

2. (Currently Amended) The system recited in Claim 1 wherein the SMI handling code reads the status latch, and if the status latch is zero, exits the SMI handling code, and if the status latch is non-zero, writes 0A to I/O location 0x70, reads I/O location 0x71, and if bit 7 of the value read from I/O location 0x71 is set, repeats the previous two steps until the value of bit 7 is not set, and then exits the SMI handling code.

3. (Currently Amended) A method for improving system management interrupt (SMI) latency of a system ~~having timing sensitive registers, a status latch, an a plurality of I/O location, the method~~ comprising the steps of:

~~(1)~~ reading ~~the~~ a status latch;

~~(2)~~ stopping if the value read from the status latch is zero;

~~(3)~~ writing to a first I/O location;

~~(4)~~ reading a second I/O location; and

~~(5)~~ if a predetermined bit of the value that is read from the second I/O location is set, repeating the ~~steps (3)~~ writing and ~~(4)~~ reading steps until the predetermined bit of the read value is not set~~; and~~

~~(6) stopping the method~~.

4. (Original) The method recited in Claim 3 which reads the status latch, and stops if the status latch is zero, and if the status latch is non-zero, writes 0A to I/O location 0x70, reads I/O location 0x71, and if bit 7 of the value read from I/O location 0x71 is set, repeats the previous two steps until the value of bit 7 is not set, and then stops.

5. (Currently Amended) Software for improving system management interrupt (SMI) latency of a system having timing-sensitive registers, a status latch, and a plurality of I/O locations, comprising:

(1) a code segment that reads the status latch;

(2) a code segment that exits if the value read from the status latch is zero;

(3) a code segment that writes to a first I/O location;

(4) a code segment that reads from a second I/O location;

(5) a code segment that, if a predetermined bit of the value that is read from the second I/O location is set, repeats steps (3) and (4) until the predetermined bit of the read value is not set to zero; and

(6) a code segment that exits when the predetermined bit of the read value is not set to zero.

6. (Currently Amended) The software recited in Claim ~~2~~ 5 which reads the status latch, and stops if the status latch is zero, and if the status latch is non-zero, writes 0A to I/O location 0x70, reads I/O location 0x71, and if bit 7 of the value read from I/O location 0x71 is set, repeats the previous two steps until the value of bit 7 is not set, and then exits.